Scalable low-complexity B-spline discretewavelet transform architecture

Scalable low complexity B-spline Discrete Wavelet Transform architecture

Maurizio Martina, Guido Masera, Gianluca Piccinini

Abstract

This work presents a scalable Discrete Wavelet Transform architecture based on the B-spline factorization. In particular, we show that several wavelet filters of practical interest have a common structure in the distributed part of their B-spline factorization. This common structure is effectively exploited to achieve scalability and to save multipliers compared with a direct polyphase B-spline implementation. Since the proposed solution is more robust to coefficient quantization than direct polyphase B-spline, it features further complexity reduction. Synthesis results are reported for a 130 nm CMOS technology to enable accurate comparison with other implementations. Moreover the performance of the new wavelet transform architecture, integrated in a complete JPEG2000 model, have been collected for several images.

I. Introduction

Filter bank (FB) [1] and lifting scheme (LS) [2], along with its flipping structure (FS) form [3], are the most common solutions to implement the discrete wavelet transform (DWT). A novel approach to design DWT architectures, based on the B-spline (BS) factorization, is proposed in [4] to reduce the number of required multipliers. As detailed in [4], the gate count for the BS architecture of the 9/7, the 6/10 and the 10/18 filters is significantly reduced compared with the corresponding FB or LS implementations. In this work, we propose a new BS architecture that offers scalability and complexity advantages with respect to solution given in [4].

The BS approach is based on factorizing each DWT as

$$H(z) = z^{\delta_H} \cdot H_{BS}(z) \cdot \hat{Q}(z) \cdot h_0 \quad (1)$$

$$G(z) = z^{\delta_G} \cdot G_{BS}(z) \cdot \hat{R}(z) \cdot g_0 \quad (2)$$

where $H(z)$ and $G(z)$ are the Z-domain representations of the analysis low-pass and high-pass filters respectively, $H_{BS}(z) = [(1+z^{-1})/2]^{\gamma_H}$ and $G_{BS}(z) = [(1-z^{-1})/2]^{\gamma_G}$ are the BS terms, z^{δ_H} and z^{δ_G} are delay terms to model the filter memory; in (1) and (2),

$$\begin{aligned} Q(z) &= \hat{Q}(z) \cdot h_0 \quad (3) \\ &= Q_0 + Q_1(z + z^{-1}) + \ldots + Q_{N_Q}(z^{N_Q} + z^{-N_Q}) \end{aligned}$$

Corrsponding author: Maurizio MARTINA, Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino, Italy, tel: +39 011 564 4205, fax: +39 011 4217, email: maurizio.martina@polito.it

The authors are with CERCOM (Center for Multimedia Radio Communications) - Dipartimento di Elettronica - Politecnico di Torino.

$$R(z) = \hat{R}(z) \cdot g_0 \quad (4)$$
$$= R_0 + R_1(z + z^{-1}) + \ldots + R_{N_R}(z^{N_R} + z^{-N_R})$$

are referred to as the filter distributed part. In (1) and (2) $H_{BS}(z)$ and $G_{BS}(z)$ account for the γ_H and γ_G zeros of $H(z)$ and $G(z)$ in z=-1 and z=1 respectively. As pointed out in [4], direct polyphase implementation of $H_{BS}(z)$ and $G_{BS}(z)$, obtained by cascading γ_H (γ_G) multiplierless stages (see Fig. 2 (a)), is preferred to the Pascal expression for long-tap filters.

On the other hand, the implementation of the distributed part, (3) and (4), requires multiplications [4]. Several works in the literature address the multiplierless implementation of the DWT. As an example [5], [6], [7] deal with FB DWT, [5], [8], [9] with LS/FS DWT and [10] with BS DWT. In particular in [10], the use of Canonic Signed Digit representation is proposed to reduce the distributed terms complexity in BS based architectures. However, only [4] and [10] investigate BS architectures, that, as shown in [4], feature a reduced number of multipliers compared with FB and LS approaches. Moreover, none of the solutions proposed in the literature exploits the algebraic properties of the distributed part to further reduce the complexity of the DWT. As a first step, this work shows, in section II, that the distributed part has a common processing structure. Consequently, the scientific contribution of this work is to detail how this structure allows for (i) lower number of multiplications, (ii) scalability, (iii) robustness to coefficient quantization with respect to direct polyphase BS implementation. These three aspects are detailed in section III and IV. In particular, in section IV, the robustness to coefficient quantization is proved by showing experimental results obtained integrating the proposed solution into JPEG2000, the latest international image compression standard, verification model [11].

II. Proposed architecture

As proved in [12], several DWT filters of practical interest in image compression are obtained from

$$H(\xi)\tilde{H}(\xi) = [\cos(\xi/2)]^{2l} \cdot \Phi_{l-1}(\theta) \quad (5)$$
$$= [\cos(\xi/2)]^{2l} \cdot \sum_{i=0}^{l-1} \begin{pmatrix} l-1+i \\ i \end{pmatrix} \theta^i$$

where $\tilde{H}(\xi)$ is the low-pass synthesis filter ($G(z)$=$\tilde{H}(-z)$ and z=$e^{j\xi}$), $2l$=γ_H+γ_G and $\theta = [\sin(\xi/2)]^2$. We obtain (1) and (2) from (5) by using the following factorization

$$[\cos(\xi/2)]^{2l} = z^{\delta_H} \cdot z^{\delta_G} \cdot H_{BS}(z) \cdot G_{BS}(-z) \quad (6)$$

$$\Phi_{l-1}(\theta) = Q(z) \cdot R(-z) \quad (7)$$

Significant examples of the filters derived from (5) are the ones considered in [4], namely the 9/7, the 6/10 and the 10/18. These filters are obtained by proper spectral factorization with $2l$=8 for the 9/7 and the 6/10, and $2l$=14 for the 10/18.

Since $\Phi_{l-1}(\theta)$ is a polynomial with real coefficients its roots are real (r) and complex conjugate pairs (c, c^*). We can then write $Q(z)$ and $R(z)$ in the form

$$Q(z) = \prod_{r\in\mathcal{I}_r^Q} L_r(z) \cdot \prod_{a,b\in\mathcal{I}_{a,b}^Q} W_{a,b}(z) \tag{8}$$

$$R(z) = \prod_{r\in\mathcal{I}_r^R} L_r(-z) \cdot \prod_{a,b\in\mathcal{I}_{a,b}^R} W_{a,b}(-z) \tag{9}$$

where $L_r(z)$ and $W_{a,b}(z)$ are

$$L_r(z) = \alpha_0 + \alpha_1(z + z^{-1}) \tag{10}$$

$$\alpha_0 = 1 - \frac{1}{2r} \qquad \alpha_1 = \frac{1}{4r} \tag{11}$$

$$W_{a,b}(z) = \beta_0 + \beta_1(z + z^{-1}) + \beta_2(z^2 + z^{-2}) \tag{12}$$

$$\beta_0 = 1 - \frac{b}{2a} + \frac{3}{8a} \quad \beta_1 = \frac{b-1}{4a} \quad \beta_2 = \frac{1}{16a} \tag{13}$$

with a=$c \cdot c^*$, b=c+c^* and $\mathcal{I}_r^Q$ ($\mathcal{I}_r^R$) and $\mathcal{I}_{a,b}^Q$ ($\mathcal{I}_{a,b}^R$) are the sets of real and complex conjugate roots in $Q(z)$ ($R(z)$).

Implementation of each $L_r(z)$ ($W_{a,b}(z)$) filter requires two (three) multiplications for α_0, α_1 in (10) (for β_0, β_1, β_2 in (12)). The number of multiplications can be reduced by formulating the filtering operation in the following matricial form. Said $x[n]$ a discrete-time input signal, output of filter $L_r(z)$ and $W_{a,b}(z)$ are computed as

$$y_L[n] = \begin{bmatrix} 1 \\ 1/r \end{bmatrix}^t \cdot \begin{pmatrix} 1 & 0 \\ -1/2 & 1/4 \end{pmatrix} \cdot \begin{bmatrix} p[0] \\ p[1] \end{bmatrix} \tag{14}$$

$$y_W[n] = \begin{bmatrix} 1 \\ b/a \\ 1/a \end{bmatrix}^t \cdot \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1/4 & 0 \\ 3/8 & -1/4 & 1/16 \end{pmatrix} \cdot \begin{bmatrix} p[0] \\ p[1] \\ p[2] \end{bmatrix} \tag{15}$$

where $[\cdot]^t$ means array transposition, $p[0]$=$x[n]$, $p[1]$=$x[n-1]$+$x[n+1]$ and $p[2]$=$x[n-2]$+$x[n+2]$. The implementation of (10) and (12) requires five multipliers, whereas (14) and (15) can be implemented as shown in Fig. 1 (a) and 1 (b), with a total of three multipliers. Low-pass and high-pass results are obtained by selectively adding or subtracting odd power terms in $L_r(z)$ and $W_{a,b}(z)$ (lp/hp signal in Fig. 1). Furthermore, Fig. 1 (c) shows that both $L_r(z)$ and $W_{a,b}(z)$ can be implemented as a single module ($LW(z)$) resorting to two multiplexers, driven by the LW signal. However, since the BS terms are in polyphase form and the distributed part is in not-polyphase form, as shown in Fig. 2, we need to properly connect BS term outputs, x_e (x'_e) and x_o (x'_o), to the distributed part input by means of registers (see Fig. 1 (d)). Moreover, registers are required when more $L_r(z)$ or $W_{a,b}(z)$ stages are cascaded to implement $Q(z)$ and $R(z)$, as in the case of the 10/18 filters, where the ouput of the first stage ($\tilde{x}$) becomes the input of the second stage (see Fig. 1 (e) and Fig. 2 (b)).

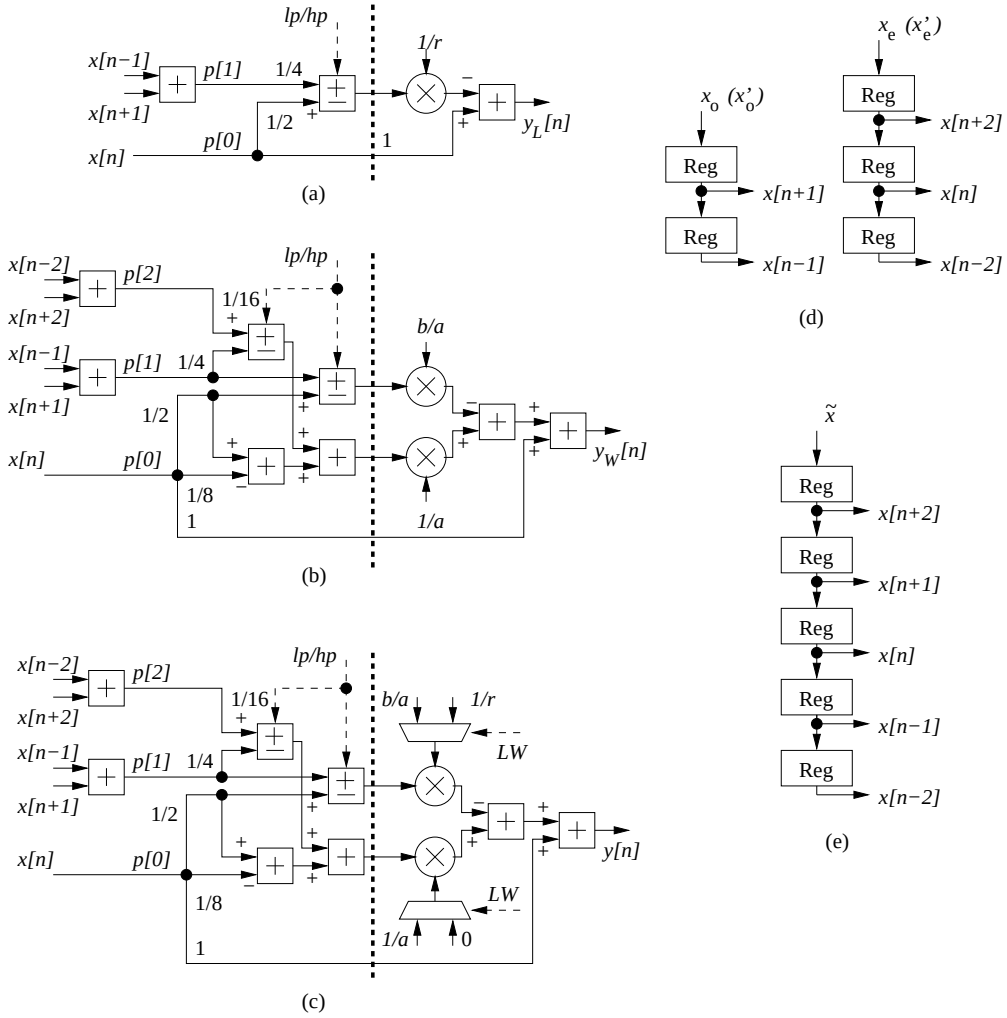

Figure 1. Block scheme of $L_r(z)$ (a), $W_{a,b}(z)$ (b) and flexible $LW(z)$ (c)

III. Results

In this work we analyze the filters considered in [4]: the 9/7, 6/10 and 10/18 wavelet filters, whose BS part is completely described by (γ_H^f, γ_G^f) with $f \in \mathcal{J}_f$=$\{9/7, 6/10, 10/18\}$, namely ($\gamma_H^{9/7}$=4, $\gamma_G^{9/7}$=4), ($\gamma_H^{6/10}$=3, $\gamma_G^{6/10}$=5) and ($\gamma_H^{10/18}$=5, $\gamma_G^{10/18}$=9). The 9/7 and 6/10 wavelet filters derive from (5) with $2l$=8, and

$$\Phi_3(\theta) = 1 + 4\theta + 10\theta^2 + 20\theta^3 = 0 \tag{16}$$

$\Phi_3(\theta)$ has only a real root r and a pair of complex conjugate roots c, c^* that lead to

$$Q^{9/7}(z) \quad = \quad W_{a,b}(z) \tag{17}$$

$$Q^{6/10}(z) \quad = \quad L_r(z) \tag{18}$$

Table I

COMPLEXITY REQUIREMENTS OF THE BS DWT ARCHITECTURE DESCRIBED IN [4] AND THE PROPOSED ONE WITH A CLOCK FREQUENCY CONSTRAIN OF 200 MHZ: MULTIPLIERS, ADDERS, REGISTERS, AND EQUIVALENT GATES. THE NUMBER OF MULTIPLIERS INCLUDES h_0 AND g_0 IN (1) AND (2)

Filter	Architecture	Multipliers	Adders	Registers	Area[(a)] [kgate] / [μm^2]	Area[(b)] [kgate] / [μm^2]
9/7 or 6/10	[4] BS Type I	5	22	16	9.80 / 58771	9.08 / 54504
	our	**3**	**27**	**16**	**7.52 / 45090**	**5.26 / 31532**
10/18	[4] BS	8	40	28	17.20 / 103195	17.20 / 103195
	our	**6**	**52**	**29**	**14.42 / 86499**	**11.27 / 67612**

[(a)] Results obtained by using 16-by-16 multipliers and 16 bit rounded output.

[(b)] Results obtained by sizing the multipliers as detailed in section IV.

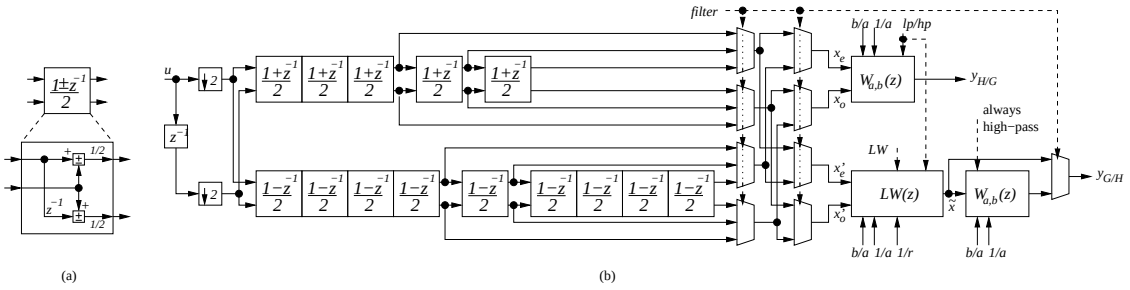

Figure 2. (a) BS basic block architecture (b) scalable BS architecture to support the 9/7, 6/10 and 10/18 wavelet filters

with

$$Q^{9/7}(z) = R^{6/10}(-z) \tag{19}$$

$$R^{9/7}(z) = Q^{6/10}(-z) \tag{20}$$

Since $\gamma_H^{9/7}$+$\gamma_G^{9/7}$=$\gamma_H^{6/10}$+$\gamma_G^{6/10}$ we can infer that the 9/7 and 6/10 architectures have the same complexity. On the other hand, the 10/18 wavelet filters are obtained from (5) with $2l$=14 and

$$\Phi_6(\theta) = 1+7\theta+28\theta^2+84\theta^3+210\theta^4+462\theta^5+924\theta^6 = 0 \tag{21}$$

whose solution is three pairs of complex conjugate roots. Said c_0, c_0^* and c_2, c_2^* the couples with minimum and maximum modulus, we obtain

$$Q^{10/18}(z) = W_{a_1,b_1}(z) \tag{22}$$

$$R^{10/18}(z) = W_{a_0,b_0}(-z) \cdot W_{a_2,b_2}(-z) \tag{23}$$

where a_i=$c_i \cdot c_i^*$ and b_i=c_i+c_i^*.

To prove the effectiveness of our methodology we described in VHDL both the BS architectures detailed in [4] and the proposed ones and synthesized them on a 0.13 μm standard cell technology with Synopsys Design Compiler. The architecture bit-width is the same employed in [4], namely internal bit-widths are all 16 bit and

Table II
COMPLEXITY REQUIREMENTS OF THE BS DWT ARCHITECTURE DESCRIBED IN [4] AND THE PROPOSED ONE CONSTRAINING THE AREA TO BE MINIMAL: EQUIVALENT GATES AND CRITICAL PATH

Filter	Architecture	Area[(a)] [kgate] / [μm^2]	Critical path[(a)] [ns]	Area[(b)] [kgate] / [μm^2]	Critical path[(b)] [ns]
9/7 or 6/10	[4] BS Type I	8.21 / 49264	8.83	7.61 / 45647	7.77
	our	**6.59 / 39529**	**6.36**	**5.00 / 30034**	**4.94**
10/18	[4] BS	13.73 / 82382	9.08	13.73 / 82382	9.08
	our	**12.59 / 75566**	**6.36**	**10.26 / 61544**	**5.28**

[(a)] Results obtained by using 16-by-16 multipliers and 16 bit rounded output.

[(b)] Results obtained by sizing the multipliers as detailed in section IV.

16-by-16 multipliers with 16 bit rounded output are used. Basic block complexity, estimated after logical synthesis, is about 1500, 70 and 90 equivalent gates for a 16-by-16 multiplier, a 16 bit adder and a 16 bit register respectively. It is worth pointing out that these values are obtained by synthesizing the basic blocks as stand-alone components, whereas the gate count for the whole BS DWT architectures are obtained by fixing the target clock frequency and enabling the optimization options of the logic synthesizer. As detailed in Table I the proposed methodology compared with [4] reduces the number of multipliers, while slightly increasing the number of adders and keeping the same number of registers for 9/7 and 6/10 filters and nearly the same for 10/18 filters. The gate count complexity for the whole BS DWT architectures synthesized with a 200 MHz clock frequency is given in the sixth column of Table I. It is worth pointing out that the complexity figures detailed in Table I include h_0, g_0 products in (1), (2), whereas these products are not considered in [4] (Tables I, II, III, IV).

In order better highlight the critical path and timing of the proposed architecture we performed also logic synthesis constraining the area to be minimal and leaving to the synthesized the burden of finding the best possible clock period. This new set of results, shown in the third and fourth columns of Table II, strengthens the effectiveness of the proposed architecture in reducing not only the complexity but also the critical path.

Finally, to prove the scalability of the proposed approach we implemented two architectures that support the on-line switching among the 9/7, 6/10 and 10/18 filters. Both the architectures require multiplexers in the BS part to support the aforementioned filters. As far as the distributed part is concerned, the first architecture is derived from the BS solution in [4]: it supports $Q^{10/18}(z)$ and $R^{10/18}(z)$, shorter filters are obtained by setting unused taps to zero. The second architecture, depicted in Fig. 2 (b), is based on the proposed approach and employs two $W_{a,b}(z)$ modules and the flexible $LW(z)$ module shown in Fig. 1 (b) and Fig. 1 (c) respectively to produce low-pass (y_H) and high-pass (y_G) results from the input signal (u). The proper $1/r$, b/a and $1/a$ values are chosen according to the selected filters. Post synthesis results for a 200 MHz clock frequency confirm the effectiveness of the proposed solution: the architecture derived from the BS solution in [4] requires 17.34 equivalent kgates, whereas the proposed one requires only 15.54 equivalent kgates.

Table III
DISTRIBUTED PART COEFFICIENTS

filter	Q_i, R_i	$1/r$, b/a, $1/a$
9/7 (6/10)	Q_0= 4.10753250160977	b/a =-1.079303580344
	Q_1=-1.98174636937784	$1/a$ = 6.847681897167
	Q_2= 0.42798011857296	
	R_0= 2.460348209828	$1/r$ =-2.920696419656
	R_1=-0.730174104914	
10/18	Q_0= 6.21914113482665	b_1/a_1=-2.603974030008
	Q_1=-3.26242958738378	$1/a_1$ =10.445744319527
	Q_2= 0.65285901997045	
	R_0= 36.6061201376705	b_0/a_0=-6.457178409811
	R_1=-27.1736134996998	$1/a_0$ =12.114739453982
	R_2= 12.1358244089364	b_2/a_2= 2.061152439819
	R_3=-3.11080643151506	$1/a_2$ = 7.301607799117
	R_4= 0.34553545344324	

IV. QUANTIZATION OF FILTER COEFFICIENTS

Further complexity can be saved by choosing the proper number of bits to represent filter coefficients. To this purpose the proposed solution was integrated into the lossy convolution-based mode of the JPEG2000 verification model [11]. Experimental simulations were performed on five standard images, namely 'Lenna' 256×256 (img1), 'Boat' 512×512 (img2), 'Goldhill' 512×512 (img3), 'Barbara' 512×512 (img4) and 'Fingerprint' 512×512 (img5). The number of DWT decomposition levels (L) has been varied from 1 to 3 for the 256×256 image and from 1 to 4 for 512×512 images ($L \in \mathcal{J}_L$=$[1,4]$). Several bit-rates (ρ) have been tested with $\rho \in \mathcal{J}_\rho$=$\{0.125, 0.25, 1, 2, 4, 8\}$ bit per pixel (bpp) with the default JPEG2000 SNR progressive mode. The other encoding parameters have been left to their default values. We performed simulations quantizing only the distributed part of the wavelet filters. Floating point values of the distributed part are summarized in Table III. Let's consider Q_i, R_i and $1/r$, b/a, $1/a$ as two complement values with k bits to represent the fractional part. First we performed a floating point simulation to obtain the performance bounds of the 9/7, 6/10 and 10/18 filters with the default JPEG2000 lossy compression mode. Then varying k from 16 down to 0 we obtained several sets of peak signal to noise ratio (PSNR) values. We indicate each set as $\text{PSNR}_m^f(\text{img}, L, \rho, k)$, where f is the filter, $f \in \mathcal{J}_f$, m is the quantized amount $m \in \{Q_i, R_i; 1/r, b/a, 1/a\}$, img is the considered image, img $\in \mathcal{J}_{\text{img}}$={img1, img2, …, img5}, L, ρ and k are the parameters defined above. In the following we refer to the floating point simulation results as k=∞ ($\text{PSNR}_m^f(\text{img}, L, \rho, \infty)$). For each f and m we define

$$\begin{aligned}\Delta\text{PSNR}_m^f(k) \quad = \quad & \max_{\{\text{img}, L, \rho\}} \Big\{ \text{PSNR}_m^f(\text{img}, L, \rho, \infty) + \\ & - \text{PSNR}_m^f(\text{img}, L, \rho, k) \Big\} \end{aligned} \tag{24}$$

as the maximum difference between the floating point PSNR and the corresponding PSNR obtained with a certain k value. In Fig. 3 $\Delta\text{PSNR}_m^f(k)$ in dB is shown for the 9/7, 6/10 and 10/18 filters. The solid lines represent the values

Table IV

AVERAGE PSNR IN DB OF THE 9/7 FILTERS AT DIFFERENT BIT-RATES WITH DIFFERENT WAVELET DECOMPOSITION LEVELS FOR k=10, 8, 6, 4 (BS AS IN [4]) AND k=6, 4, 2, 0 (PROPOSED)

L	ρ=8	ρ=4	ρ=2	ρ=1	ρ=0.5	ρ=0.25	ρ=0.125	ρ=8	ρ=4	ρ=2	ρ=1	ρ=0.5	ρ=0.25	ρ=0.125
	9/7 k=10							9/7 k=8						
1	49.58	49.58	42.89	36.59	31.24	26.42	20.52	49.44	49.44	42.92	36.58	31.27	26.36	20.52
2	49.33	49.33	43.63	37.65	33.12	29.32	25.85	48.92	48.92	43.55	37.64	33.12	29.36	25.97
3	49.29	49.29	43.73	37.86	33.45	29.81	26.96	48.48	48.48	43.57	37.85	33.46	29.85	26.98
4	49.30	49.30	43.35	37.59	33.40	30.06	27.32	48.25	48.25	43.11	37.57	33.39	30.08	27.34
	9/7 k=6							9/7 k=4						
1	48.20	48.20	42.44	36.42	31.09	26.26	20.52	47.36	47.36	42.08	36.21	30.78	26.11	20.58
2	45.19	45.19	41.90	37.13	32.87	29.17	25.84	43.64	43.64	41.05	36.78	32.70	29.03	25.90
3	42.55	42.55	40.52	36.70	32.93	29.60	26.87	40.74	40.74	39.27	36.12	32.67	29.46	26.79
4	41.36	41.36	39.56	36.17	32.73	29.70	27.17	39.46	39.46	38.20	35.46	32.38	29.49	27.09
	Proposed 9/7 k=6							Proposed 9/7 k=4						
1	49.58	49.58	42.90	36.58	31.27	26.42	20.51	49.59	49.59	42.92	36.58	31.21	26.40	20.55
2	49.33	49.33	43.64	37.67	33.09	29.32	25.85	49.32	49.32	43.61	37.65	33.10	29.32	25.89
3	49.29	49.29	43.76	37.86	33.43	29.82	26.95	49.28	49.28	43.73	37.87	33.44	29.88	26.96
4	49.31	49.31	43.35	37.62	33.40	30.06	27.33	49.29	49.29	43.34	37.59	33.39	30.06	27.33
	Proposed 9/7 k=2							Proposed 9/7 k=0						
1	49.54	49.54	42.90	36.56	31.21	26.30	20.57	49.51	49.51	42.84	36.55	31.10	26.28	20.54
2	49.22	49.22	43.61	37.63	33.12	29.33	25.94	49.02	49.02	43.48	37.60	33.08	29.27	25.91
3	49.10	49.10	43.71	37.86	33.46	29.86	26.94	48.64	48.64	43.49	37.77	33.39	29.87	26.90
4	49.05	49.05	43.30	37.60	33.41	30.08	27.35	48.44	48.44	43.05	37.50	33.34	30.02	27.27

obtained by quantizing Q_i and R_i, whereas the dashed lines detail the values achieved quantizing $1/r$, b/a, $1/a$. As it can be observed, the curves referred to the 9/7 and 6/10 filters are nearly overlapped. Since representing Q_i and R_i with $k < 2$ causes $H(z)$ and $G(z)$ to degenerate to band pass filters, solid line simulations have been carried out for $k \in [2, 16]$. Conversely, the proposed solution with k=0 (only integer part of $1/r$, b/a, $1/a$) introduces a maximum PSNR degradation of about 1 dB for the 9/7 and 6/10 filters and of about 3.5 dB for the 10/18 filters. As it can be inferred from Fig. 3, when $k < 10$ the quantization of Q_i and R_i leads to significant performance loss. On the other hand, the quantization of $1/r$, b/a, $1/a$ worsens the PSNR when $k < 6$.

In Table IV we show for the 9/7 filters the PSNR obtained by averaging the mean square error values achieved for the five test images belonging to $\mathcal{J}_{\text{img}}$. The simulation parameters have been changed in the following ranges: $L \in \mathcal{J}_L$, $\rho \in \mathcal{J}_\rho$, $k \in [4, 10]$ for Q_i, R_i and $k \in [0, 6]$ for $1/r$, b/a, $1/a$. The quantization of Q_i and R_i leads to significant PSNR degradation mainly for ρ=1 bpp or higher when $k \leq 8$ (ΔPSNR$\geq$1.2dB). On the contrary, the proposed solution keeps the PSNR degradation limited to less than 0.5 dB with k=4. Similarly in Table V and VI we show the results obtained for the 6/10 and 10/18 filters respectively, using the same setup employed for the 9/7 filters. As it can be observed the proposed approach leads to excellent results also with the 6/10 and 10/18 wavelet filters.

Table V

AVERAGE PSNR IN DB OF THE 6/10 FILTERS AT DIFFERENT BIT-RATES WITH DIFFERENT WAVELET DECOMPOSITION LEVELS FOR k=10, 8, 6, 4 (BS AS IN [4]) AND k=6, 4, 2, 0 (PROPOSED)

L	ρ=8	ρ=4	ρ=2	ρ=1	ρ=0.5	ρ=0.25	ρ=0.125	ρ=8	ρ=4	ρ=2	ρ=1	ρ=0.5	ρ=0.25	ρ=0.125
	6/10 k=10							6/10 k=8						
1	49.56	49.56	43.15	36.86	31.61	26.16	19.10	49.41	49.41	43.14	36.87	31.66	26.31	19.08
2	49.30	49.30	43.71	37.70	33.21	29.44	26.21	48.90	48.90	43.64	37.70	33.23	29.46	26.23
3	49.27	49.27	43.76	37.87	33.47	29.86	27.06	48.47	48.47	43.60	37.86	33.47	29.90	27.02
4	49.27	49.27	43.38	37.61	33.41	30.04	27.30	48.23	48.23	43.14	37.56	33.37	30.03	27.32
	6/10 k=6							6/10 k=4						
1	48.19	48.19	42.64	36.70	31.48	26.21	19.10	47.34	47.34	42.36	36.61	31.47	26.03	18.82
2	45.19	45.19	41.98	37.18	32.97	29.35	26.03	43.64	43.64	41.11	36.88	32.80	29.20	25.88
3	42.55	42.55	40.56	36.74	32.93	29.66	26.82	40.74	40.74	39.28	36.11	32.62	29.45	26.82
4	41.35	41.35	39.59	36.18	32.73	29.69	27.13	39.46	39.46	38.19	35.45	32.34	29.46	27.03
	Proposed 6/10 k=6							Proposed 6/10 k=4						
1	49.55	49.55	43.15	36.86	31.62	26.32	19.09	49.56	49.56	43.14	36.85	31.60	26.19	18.91
2	49.30	49.30	43.73	37.71	33.20	29.44	26.21	49.31	49.31	43.71	37.70	33.21	29.44	26.21
3	49.27	49.27	43.78	37.87	33.48	29.85	27.05	49.26	49.26	43.77	37.87	33.47	29.91	26.95
4	49.29	49.29	43.40	37.60	33.41	30.05	27.31	49.27	49.27	43.35	37.58	33.40	30.03	27.31
	Proposed 6/10 k=2							Proposed 6/10 k=0						
1	49.50	49.50	43.13	36.89	31.64	26.26	18.94	49.49	49.49	43.08	36.83	31.62	26.16	18.88
2	49.19	49.19	43.70	37.70	33.20	29.44	26.15	49.00	49.00	43.56	37.67	33.20	29.43	26.16
3	49.06	49.06	43.76	37.85	33.48	29.88	27.03	48.63	48.63	43.51	37.78	33.43	29.90	26.85
4	49.01	49.01	43.34	37.60	33.40	30.04	27.28	48.42	48.42	43.08	37.49	33.34	30.00	27.29

Logical synthesis results presented in section III have been obtained with 16-by-16 multipliers, 16 bit rounded output and k=12 for Q_i, R_i, $1/r$, b/a, $1/a$ in the case of 9/7 and 6/10 filters; 10/18 filters were implemented with k=9 for Q_i, R_i and k=11 for b_i/a_i, $1/a_i$. To insure limited performance degradation introduced by Q_i and R_i quantization, k=9 is adequate (Fig. 3). On the other hand, we can obtain nearly the same performance with the proposed solution and k=4. To that purpose, we performed new logical synthesis for a target clock frequency of 200 MHz using 16-by-13 multipliers (k=9) and 16-by-16 multipliers (k=9) to represent Q_i and R_i for the 9/7-6/10 and 10/18 filters respectively. Similarly, we used 16-by-8 multipliers (k=4) and 16-by-9 multipliers (k=4) for the proposed 9/7-6/10 and 10/18 architectures respectively. As shown in the seventh column of Table I the quantization robustness of the proposed solution significantly reduces the area requirement. In the fifth and sixth column of Table II, the area and the critical path obtained by constraining the area to be minimal and leaving to the synthesized the burden of finding the best possible clock period are shown. This new set of results, confirms the reduced complexity and critical path figures of the proposed architectures. Finally, the aforementioned quantization approach was used on the scalable architectures that support the on-line switching among the 9/7, 6/10 and 10/18 filters. The architecture derived from the BS solution in [4], sized on the 10/18 filters still requires 16-by-16 multipliers (k=9) leading to 17.34 equivalent kgates for a 200 MHz clock frequency. For the same clock frequency, the proposed architecture requires 16-by-9 multipliers (k=4) leading to only 13.32 equivalent kgates.

Table VI

AVERAGE PSNR IN DB OF THE 10/18 FILTERS AT DIFFERENT BIT-RATES WITH DIFFERENT WAVELET DECOMPOSITION LEVELS FOR k=10, 8, 6, 4 (BS AS IN [4]) AND k=6, 4, 2, 0 (PROPOSED)

L	ρ=8	ρ=4	ρ=2	ρ=1	ρ=0.5	ρ=0.25	ρ=0.125	ρ=8	ρ=4	ρ=2	ρ=1	ρ=0.5	ρ=0.25	ρ=0.125
	10/18 k=10							10/18 k=8						
1	49.52	49.52	43.24	37.01	31.71	26.47	19.36	49.50	49.50	43.26	37.01	31.69	26.47	19.34
2	49.23	49.23	43.83	37.85	33.43	29.54	26.18	49.20	49.20	43.81	37.88	33.42	29.48	26.19
3	49.14	49.14	43.88	38.06	33.67	30.00	27.04	49.11	49.11	43.90	38.07	33.66	29.99	27.13
4	49.11	49.11	43.46	37.77	33.61	30.21	27.41	49.09	49.09	43.46	37.79	33.60	30.20	27.40
	10/18 k=6							10/18 k=4						
1	48.77	48.77	43.16	37.03	31.72	26.48	19.25	48.29	48.29	42.87	36.92	31.68	26.30	19.18
2	47.09	47.09	43.21	37.80	33.37	29.58	26.30	46.46	46.46	42.74	37.59	33.28	29.51	26.26
3	45.36	45.36	42.50	37.70	33.58	30.04	27.07	45.33	45.33	42.26	37.55	33.48	29.96	26.99
4	44.50	44.50	41.76	37.30	33.45	30.16	27.40	45.29	45.29	41.93	37.28	33.38	30.08	27.37
	Proposed 10/18 k=6							Proposed 10/18 k=4						
1	49.54	49.54	43.25	37.00	31.71	26.47	19.36	49.55	49.55	43.23	37.01	31.67	26.49	19.34
2	49.29	49.29	43.84	37.87	33.43	29.52	26.11	49.29	49.29	43.82	37.87	33.43	29.53	26.12
3	49.26	49.26	43.90	38.06	33.68	29.99	27.04	49.25	49.25	43.90	38.06	33.67	30.01	27.03
4	49.27	49.27	43.48	37.78	33.61	30.18	27.40	49.25	49.25	43.48	37.76	33.60	30.19	27.41
	Proposed 10/18 k=2							Proposed 10/18 k=0						
1	49.54	49.54	43.22	36.99	31.68	26.23	19.27	48.78	48.78	43.15	37.01	31.82	26.39	19.09
2	49.26	49.26	43.80	37.92	33.40	29.51	26.25	47.68	47.68	43.44	37.83	33.42	29.57	26.28
3	49.17	49.17	43.87	38.03	33.65	30.00	27.06	46.71	46.71	43.13	37.84	33.63	30.02	27.09
4	49.15	49.15	43.44	37.76	33.60	30.18	27.39	46.21	46.21	42.55	37.54	33.53	30.17	27.33

V. CONCLUSION

In this work we propose a scalable BS DWT architecture that employs a reduced number of multipliers. Implementation results on a 0.13 μm standard cell technology prove the complexity reduction offered by the proposed methodology. Finally, simulations into a JPEG2000 model show that the proposed methodology is very robust to filter coefficients quantization leading to further complexity reduction.

REFERENCES

[1] G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge, MA: Wellesley, 1996.

[2] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247–269, 1998.

[3] C. T. Huang, P. C. Tseng, and L. G. Chen, "Flipping Structure: an efficient VLSI architecture for lifting-based discrete wavelet transform," *IEEE Trans. on Signal Processing*, vol. 52, no. 4, pp. 1080–1089, Apr. 2004.

[4] ——, "VLSI architecture for forward discrete wavelet transform based on B-spline factorization," *Journal of VLSI Signal Processing*, vol. 40, pp. 343–353, 2005.

[5] K. A. Kotteri, S. Barua, A. E. Bell, and J. E. Carletta, "A comparison of hardware implementations of the biorthogonal 9/7 DWT: convolution versus lifting," *IEEE Trans. on Circuits and Systems II*, vol. 52, no. 5, pp. 256–260, May 2005.

[6] X. Cao, Q. Xie, C. Peng, Q. Wang, and D. Yu, "An efficient VLSI implementation of distributed architecture for DWT," in *IEEE Workshop on Multimedia Signal Processing*, 2006, pp. 364–367.

[7] P. Longa, A. Miri, and M. Bolic, "Modified distributed arithmetic based architecture for discrete wavelet transforms," *IET Electronics Letters*, vol. 44, no. 4, pp. 270–271, Feb. 2008.

[8] D. Tay, "A class of lifting based integer wavelet transform," in *IEEE International Conference on Image Processing*, 2001, pp. 602–605.

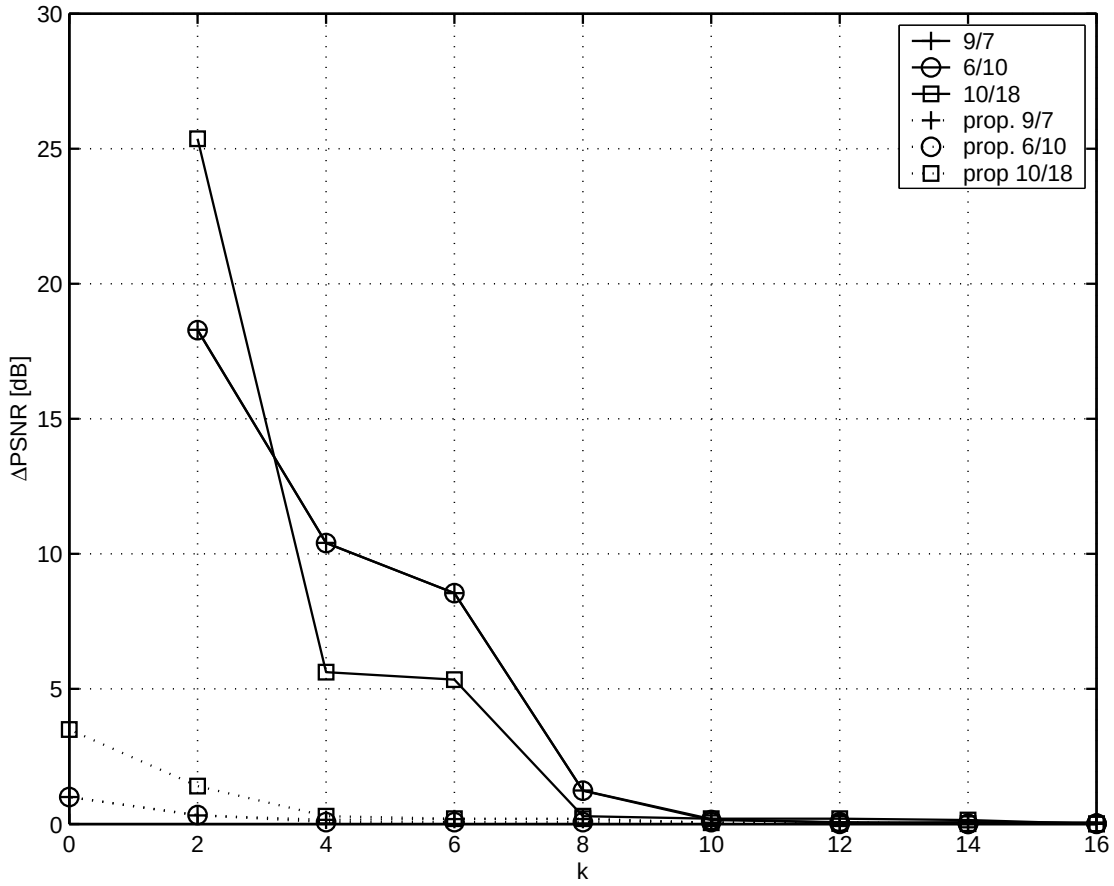

Figure 3. Maximum difference between floating and fixed point PSNR (ΔPSNR) versus the number of bits used to represent the fractional part (k). Solid lines refer to the BS implementation in [4] and dashed lines to the proposed factorization methodology

[9] M. Martina and G. Masera, "Folded multiplierless lifting-based wavelet pipeline," *IET Electronics Letters*, vol. 43, no. 5, pp. 27–28, Mar. 2007.

[10] K. A. Kotteri, A. E. Bell, and J. E. Carletta, "Multiplierless filter bank design: Structures that improve both hardware and image compression performance," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 6, pp. 776–780, Jun. 2006.

[11] M. Boliek, "JPEG 2000 Final Committee Draft," 2000.

[12] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using the wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.